



Abstract

This White Paper outlines the data structure that HighStone applies in holding Asset Survey and Routine Maintenance data records. These definitions include the type of Asset Survey that can be raised against Asset Records, and record details of Asset Condition and Cyclic Maintenance Activities.

HighStone is designed to build up and hold a complete history of the maintenance and checking of all assets held within the contract database. Additional maintenance records can be created to monitor specific aspects of an asset type and provide a complete history log from within the same data framework.

This document is intended as a technical briefing on HighStone database definitions and usage - it is not written as a User Manual for the day to day operation of HighStone.

Document Summary

Title of Document	HighStone Asset Management Definitions (Surveys)
Date of Issue	January 2012
Document Version	1.1

This document is written by Claremont Controls Ltd and made available to support users of HighStone in the use and application of the system within their organisation. Users of HighStone are reminded that these notes cover HighStone as a whole and hence details given may include reference to elements that are not used or not available on their particular application configurations.

While every reasonable precaution has been taken in the preparation of this document, Claremont Controls Limited does not assume responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

The information contained in this document is believed to be accurate at the time of drafting but it is strongly recommended that readers check that the details given are still appropriate before committing to significant configuration work based on the topics presented herein. No guarantee is provided and the document is provided on an 'as-is' basis. Readers use this information entirely at their own risk.

Claremont Controls Limited
Suite 4, Wansbeck Business Centre
Rotary Parkway
Ashington
Northumberland, United Kingdom
NE63 8QZ

UK Tel: (01670) 819000
Int Tel: +44 1670 819000
Web: www.claremontcontrols.co.uk
Email: info@claremontcontrols.co.uk

© 2012, Claremont Controls Limited. All rights reserved.



Contract Surveys Definitions

Cyclic and routine maintenance tasks and asset inspections are recorded within the Contract Database under records of Asset Surveys that are undertaken. The data structures held are defined as follows:

- Survey Header Records - Defines the class of survey and reason why the survey was raised.
- Network sections that have been included within the Survey - including details as to when each section was surveyed, and who undertook the survey.
- Record of Asset Defects, and Asset Condition assessments that have been identified as part of the survey.

In the field Survey Records may be recorded on hand-held mobile devices offering full data integration with the main Contract Database, or recorded on paper or other systems and the data returned as a separate exercise after the survey has been completed.

Contract Asset Surveys clearly carry close links to the Asset Inventory held in the Contract Database, and the Defects Records also held in the system. It should be noted that HighStone collates all Asset activities of Routine or Cyclic Maintenance, Condition Assessments and Defect (Damage) Records under the same data definition structure. This allows complete Maintenance Histories for individual assets to be built up and reported on easily by the system.

If it is required that Asset Record histories are to be maintained and held in the Contract Database, then it is essential that the Asset Records are created and maintained in a consistent manner throughout the contract term. This may appear to be a fairly obvious statement however there are some scenarios where a defined strategy is required:

- Where in the field surveys are undertaken as an independent assessment of what is found - starting from a blank sheet and recording all assets now found at the location - then when this information is uploaded in to the Contract Database the only viable approach is to 'clear all old asset records' and then 'add assets records now recorded'. This action will create entirely *new* assets in the Contract Database, and any maintenance records held against original records will not be carried forward. This is usually undesirable as often the assets in the field have not changed over the entire contract term.
- The same result will occur if Contract Asset record lists are continually uploaded / refreshed from external data sources. Historic asset maintenance records will be similarly lost.
- Where Assets are damaged beyond repair and a new unit is installed, the user should establish a consistent strategy as to whether the new item constitutes a 'replacement' of the original asset, or should be considered as a 'new' asset record. In the former case, a maintenance history will carry through against the single asset record, in the latter case a new maintenance history is started for the new record.

Contract Survey Data Definitions

Survey Header Entry

Individual Asset Surveys are raised as individual record entries in data table [tblProjectSurveys]. This table holds details of when a survey is raised, who by, the type of survey and reason why it was raised. This record *does not* hold details on when the survey was actually completed, by whom, and any details as to what was found during the survey.



Internal Identifier	Integer identifier for the record. This value is used by HighStone in all data table cross-references. The value is cast automatically when new definitions are added.
Survey Number	The survey reference number shown to the user and used to identify survey records.
Description	A full text description for the survey record.
Survey Type	Classifies the survey as to whether it is a routine survey, part of cyclic maintenance, a reactive survey (e.g. in response to accident damage) or raised for some other reason. Links to table [ltbSurveyTypes].
Survey Initiation	Classifies the survey as to the purpose and range of assets that are to be included in the survey. Links to table [ltbSurveyInit].
Raised By	Identifies the user who raised the survey record.
Date Raised	Records the date / time when the survey record was raised.
Sections Count	The record holds a count as to the number of geographic sections that are to be undertaken by the survey.
Object Record Links (to Incident, Works Order)	This table may also hold internal data links to other key Object Records within the Contract Database. These links are held through the numeric record identifier on Incident or Works Order object records. This link is optional, typically it is used to link a site inspection / survey that has been initiated in response to the asset damage resulting from a recorded incident or works order in the system.
Internal GUID	HighStone also holds a GUID identifier for each Survey header record. These values are used in Mobile Applications and when reporting spans across more than one Contract Database - where common GUID values can identify similar definitions across multiple databases.

This data table also holds details of independent hand-held or tablet devices that are used to undertake the survey and this information is used to manage the transfer of survey data between the Contract Database and each device. This functionality is handled by the HighGate facility in HighStone.

There is no 'status' condition held on the survey header record. Where Survey Records have been raised and are no longer required (e.g. raised in error) it is common practice to *delete* the survey records from the database completely. This action is preferred to ensure unwanted records are not picked up by functions that pass data to external mobile devices. Where Survey Records are removed in this fashion, it will lead to a break in the sequence of Survey Numbers across the data table.

Surveyed Sections Entries

Each raised survey will be targeted at one or more geographic locations within the contract area and this definition is held in [tblSectionSurveys]. In highways applications this is defined as a list of



HighStone White Paper
HighStone Asset Management Definitions (Surveys)

Version 1.1 | January 2012 | Page 4

Network Road Sections, or in other configurations this will be a range of buildings, floors and rooms etc. Each raised survey record must carry one or more entries in this table.

Internal Identifier	Integer identifier for the record. This value is used by HighStone to identify each record.
Parent Survey Identifier	Integer link reference back to the parent header record in [tblProjectSurveys].
Chart Section / Room Identifier	Integer link reference back to the geographic location record in [tblRooms].
Section Count	A sequential count that defines the order in which the geographic locations should be displayed. This definition may follow a defined 'route' for the survey, but does not necessarily constrain the survey to be undertaken strictly in this sequence.
Survey Inspector	Identifies the user who undertook the survey for the geographic location record.
Start / Finish Time	Records the date / time when the survey for the geographic location was started / completed. In some applications only a single date / time entry is recorded - possibly across all section records within the survey.
Status	Records the status of the survey for the current section record. This classifies the entry as: Raised Only, Not Done, Fully Complete, Partially Complete, Cancelled. Status definitions are held in [ltbSurveyStatus].
Partial Start / Finish Location	These two values are used on applications where the survey follows a linear asset - such as a highway or rail line. If a section is only partially surveyed, then commence and end points within the section are recorded here. These values are only set where the Status field is shown as 'Partially Complete' - in all other cases both values here are given as zero.
Site Conditions	For surveys undertaken in the field, site conditions prevailing at the time of the survey are recorded. HighStone records this detail here so that changes in site conditions over an extended survey can be recorded.
Data Upload / Download to mobile devices.	Details on when data relating to the section survey has been downloaded to, and uploaded from, mobile hand-held or tablet devices are held.
Defects Count	The record holds a count as to the number of Defects recorded by the user as part of the completed survey. This count shows the total number of <i>all</i> Defects records raised - this may include condition assessments, cyclic maintenance records and / or records of asset damage and defect.



Internal GUID	HighStone also holds a GUID identifier for each Survey Section record. These values are used in Mobile Applications and when reporting spans across more than one Contract Database - where common GUID values can identify similar definitions across multiple databases.
---------------	--

HighStone maintains details of survey start / completion details against each geographic section record in the survey in order to record 'null return' surveys. This ensures that any surveys undertaken that do not record any notifiable 'defects' are recorded within the Contract Database.

Surveys that are raised in direct response to reported Asset Damage or Defect will usually carry only one geographic section record against the survey header record.

Surveys that are raised as part of routine inspections and maintenance will include a list of custom selected geographic section records. These may or may not reflect the sequence or route that the survey itself is undertaken.

Regular survey inspections that follow the same route through the Contract Area should be defined as Survey Routes within the Contract Database so that the entire survey record may be raised quickly and accurately whenever it is required.

This data table may be used directly to provide reports listing the survey / inspection history for the geographic areas across the contract area.

Asset Defects Records

The record of the survey inspection is held as a series of assessment or defects records within the data table [tblDefects]. The table will hold a range of maintenance, assessment and defect details:

- Asset Condition Assessments will typically be recorded as a definition of 'As New', 'Satisfactory', 'Poor' or 'Failed'. Under this regime the condition must be assigned to a specific target asset record, and each survey should include assessments for *all* assets of the relevant type recorded within each geographic location. Where assets require separate checks for mechanical and, say, electrical aspects then separate assessment records will be required for each class of inspection.
- Asset Defects will be recorded as the specific reason behind the defect for any given asset. It is usual to link each recorded Defect record to a specific Asset record - although this is not a mandatory requirement in HighStone. If this link to a specific Asset record is *not* made, then a complete maintenance history reported against specific assets will not be possible.
- Cyclic maintenance and inspection tasks will be recorded within the same definition structure and note such tasks as cleaning, replacement of consumable parts, removal after critical damage, and re-instatement.
- Horticultural works such as grass-cutting, tree and scrub clearing, litter picking and seeping will also be recorded as individual entries within the same data table.

Listing Surveys by Geographic Location

Getting a list of records detailing Completed Surveys by geographic section is relatively easy. This query lists all sections that were subjected to a 'Safety' inspection, where the record records 'Completed' against the entry.

```
Select tblRooms.RoomNumber, tblProjectSurveys.PSSurveyNo,  
lrbSurveyTypes.SurveyTypeTag, lrbSurveyInit.SurveyInitTag,
```

**HighStone Asset Management Definitions (Surveys)**

```
tblSectionSurveys.SSFinishTime, ltbSurveyStatus.SurveyStatusDescrip
From tblProjectSurveys Inner Join
    tblSectionSurveys On tblProjectSurveys.SurveyId = tblSectionSurveys.SSSurveyId
Inner Join
    tblRooms On tblSectionSurveys.SSSectionId = tblRooms.RoomId Left Join
    ltbSurveyStatus On tblSectionSurveys.SSStatus = ltbSurveyStatus.SurveyStatusId
Left Join
    ltbSurveyInit On tblProjectSurveys.PSSurveyInitiated =
        ltbSurveyInit.SurveyInitId Left Join
        ltbSurveyTypes On tblProjectSurveys.PSSurveyType = ltbSurveyTypes.SurveyTypeId
Where ltbSurveyInit.SurveyInitId = 2 And ltbSurveyStatus.SurveyStatusId = 2
Order By tblRooms.RoomNumber, tblSectionSurveys.SSFinishTime Desc;
```

This selection lists *all* Safety Surveys recorded against each geographic chart section - listing each survey in reverse date order within each section, and then each section in order.

It is often the case that what is required is the *last* survey undertaken within each geographic section - thereby giving an indication as to when the next survey should be completed. This could be achieved by returning a MAX() function against the Survey Completion Date (SSFinishTime), but this will not return the Survey Reference Number for the particular survey.

The following SQL script expands the selection to return just the *latest* survey record within each geographic section. The SQL uses a Row_Number() function to set a row count within each section of Chart Section records. The outer selection in the query selects those record rows set with a count of 1 (one). Thereby the resulting recordset returns the first entry in the sub-sequence - which is the last survey undertaken within each geographic section.

```
Select RoomNumber, PSSurveyNo,
    SurveyTypeTag, SurveyInitTag,
    SSFinishTime, SurveyStatusDescrip
From
(
Select tblRooms.RoomNumber, tblProjectSurveys.PSSurveyNo,
    ltbSurveyTypes.SurveyTypeTag, ltbSurveyInit.SurveyInitTag,
    tblSectionSurveys.SSFinishTime, ltbSurveyStatus.SurveyStatusDescrip,
    Row_Number() Over (Partition by tblRooms.RoomNumber Order By
tblSectionSurveys.SSFinishTime DESC) As Rn
From
tblProjectSurveys Inner Join
    tblSectionSurveys On tblProjectSurveys.SurveyId = tblSectionSurveys.SSSurveyId
Inner Join
    tblRooms On tblSectionSurveys.SSSectionId = tblRooms.RoomId Left Join
    ltbSurveyStatus On tblSectionSurveys.SSStatus = ltbSurveyStatus.SurveyStatusId
Left Join
    ltbSurveyInit On tblProjectSurveys.PSSurveyInitiated =
        ltbSurveyInit.SurveyInitId Left Join
        ltbSurveyTypes On tblProjectSurveys.PSSurveyType = ltbSurveyTypes.SurveyTypeId
Where (ltbSurveyInit.SurveyInitId = 2) And (ltbSurveyStatus.SurveyStatusId = 2) And
(tblRooms.RoomDeleted = 0)
) tTable
Where Rn = 1
Order By RoomNumber;
```

This type of reporting is appropriate for monitoring and managing the routine Surveys and Inspections across the contract area. At this stage the report is looking at surveys undertaken, and is not addressing asset record details at all.



Asset Condition Surveys (Point Assets)

This class of survey records a specific entry in [tblDefects] for each Asset that is assessed as part of the survey - it is expected that all Assets of the required types are assessed and recorded.

The usual entries added to [tblDefects] are as follows:

Data Values	Usage and Comment
Identifier [DFIdentifier]	Record identifier that is used internally and is shown to the user as the record identifier. The identifier value is set automatically.
Survey Section Id [DFSurveyId]	This value links the record to the parent Survey geographic section in [tblSectionSurveys]. Note this link is to the survey section, and <i>not</i> a direct link to the target section as held in [tbRooms], and <i>not</i> the survey header record identifier.
Asset Item Id [DFItemId]	Link to the target Asset record - as defined for the prime record identifier in [tblRoomsRC].
Link to Master Defect Class [DFMasterId]	This entry links the defect to the parent Defect Class as defined in [tblMasterDefects]. This record class is usually classed as the 'Condition' or 'Long Stop' asset assessment class.
Link to Defect Type record [DFClassId]	This entry links the defect to the parent Defect Type as defined in [tblMasterDefectsItems]. This record definition is usually classed as the 'Condition' or 'Long Stop' asset assessment type.
Date of Entry [DFDateRaised]	Holds the date the asset condition was assessed.
Asset Condition [DFCategory]	This entry is taken from [tblDefectPriority] and will be cast as one of the 'Condition' categories: 100 - Not Specified 101 - As New 102 - Satisfactory 103 - Poor 104 - Failed

This defines those fields that are relevant to an Asset Condition Assessment survey.

The table [tblDefects] carries a range of other data values that would normally be cast to maintain a consistency across the data table. Strictly the values are not needed as they can be picked up by the link to the Assets data table [tblRoomsRC] - however some legacy reports and displays are expecting a fully populated record.

The values involved are:



HighStone White Paper
HighStone Asset Management Definitions (Surveys)

Version 1.1 | January 2012 | Page 8

Data Values	Usage and Comment
DFSectionId	Geographic location section reference - from the Asset record, or the survey section reference.
DFChainage, DFChainageStart, DFChainageFinish	Set from Asset location details.
DFLocation	Text entry - Set to 'Null'.
DFXRefCodeId, DFXRefCode	Set from Asset location details.
DFItemCodeId, DFItemCode	Set from Asset type details.
DFAtrib1Value, FDAttib2Value, DFAtribText	Not used. Set to zero / 'Null' as appropriate.
DFNoMakeSafe, DFNoTempRep, DFNoPermRep	All set to 'True' as a Condition Defect record does not require any remedial actions.
DMadeSafe, DFTempRep, DFCCompleted	All set to 'False' as not relevant.
DFDateMadeSafe, DFDateTempRep, DFDateCompleted	All set to 'Null' as not relevant.
DFMakeSafeWOld, DFTempRepWOld, DFWOld	All set to zero as not relevant.
DFTreatId, DFActionReqd	Set to zero and 'Null' as not relevant.
DFCancelled, DFDeleted	Set to 'False'.

As for geographic Section Surveys, a SQL script to extract the Condition Assessments for Asset records is straightforward to create:

```
Select tblRoomsRC.RoomsRCId, tblRoomComponents.RCItemCode, tblRooms.RoomNumber,
    ltbXSP.XSPTagCode, tblRoomsRC.RoomsRCStChain, tblDefectPriority.DPDescription,
    tblDefects.DFDateRaised, tblProjectSurveys.PSSurveyNo
From tblRoomsRC Inner Join
    tblRoomComponents On tblRoomsRC.RoomsRCCompId = tblRoomComponents.RComponentId
Inner Join
    tblRooms On tblRoomsRC.RoomsRCRoomId = tblRooms.RoomId Inner Join
    tblDefects On tblRoomsRC.RoomsRCId = tblDefects.DFItemId Inner Join
    tblSectionSurveys On tblDefects.DFSurveyId = tblSectionSurveys.SecSurveyId
Inner Join
    tblProjectSurveys On tblSectionSurveys.SSSurveyId = tblProjectSurveys.SurveyId
Left Join
    ltbXSP On tblRoomsRC.RoomsRCXRefCodeId = ltbXSP.XSPId Inner Join
    tblDefectPriority On tblDefects.DFCategory = tblDefectPriority.DPIIdentifier
Inner Join
    tblMasterDefectItems On tblDefects.DFClassId =
        tblMasterDefectItems.MDIIdentifier
Where tblRoomComponents.RCItemCont = 0 And tblMasterDefectItems.MDICode = 'LGST'
Order By tblRoomsRC.RoomsRCId, tblDefects.DFDateRaised Desc;
```

As noted above, this report lists *all* Asset Condition Assessments. The same tactic as above can be used to access the latest Condition Assessment.



```
Select RoomsRCId, RCItemCode, RoomNumber, XSPTagCode, RoomsRCStChain, DPDescription,
DFDateRaised, PSSurveyNo
From (
Select tblRoomsRC.RoomsRCId, tblRoomComponents.RCItemCode, tblRooms.RoomNumber,
ltbXSP.XSPTagCode, tblRoomsRC.RoomsRCStChain, tblDefectPriority.DPDescription,
tblDefects.DFDateRaised, tblProjectSurveys.PSSurveyNo,
Row_Number() Over (Partition by tblRoomsRC.RoomsRCId Order By tblDefects.DFDateRaised
DESC) As Rn
From tblRoomsRC Inner Join
tblRoomComponents On tblRoomsRC.RoomsRCCompId = tblRoomComponents.RComponentId
Inner Join
tblRooms On tblRoomsRC.RoomsRCRoomId = tblRooms.RoomId Inner Join
tblDefects On tblRoomsRC.RoomsRCId = tblDefects.DFItemId Inner Join
tblSectionSurveys On tblDefects.DFSurveyId = tblSectionSurveys.SecSurveyId
Inner Join
tblProjectSurveys On tblSectionSurveys.SSSurveyId = tblProjectSurveys.SurveyId
Left Join
ltbXSP On tblRoomsRC.RoomsRCXRefCodeId = ltbXSP.XSPId Inner Join
tblDefectPriority On tblDefects.DFCategory = tblDefectPriority.DPIIdentifier
Inner Join
tblMasterDefectItems On tblDefects.DFClassId =
tblMasterDefectItems.MDIIdentifier
Where (tblRoomComponents.RCItemCont = 0) And (tblMasterDefectItems.MDICode = 'LGST')
And (tblRoomsRC.RoomsRCDeleted = 0)
) tTable
Where Rn = 1
Order By RoomsRCId, DFDateRaised Desc;
```

Asset Condition Surveys (Continuous Assets)

On highways configurations, certain assets are not point items but are continuous and run along the length of the geographic Section. For these Assets, the Condition Assessment has to be made along the length and variations in condition recorded. To hold this additional information, each assessment record includes definitions of the start and end points for each length of Asset that is of the same condition.

Data Values	Usage and Comment
DFChainageStart	Defines the Start point within the section where the condition assessment starts.
DFChainageFinish	Defines the Finish point within the section where the condition assessment starts.
DFChainage	This field id not usually used. HighStone convention is to set this value to the average between the Start and Finish values.

On highways configurations, the Asset Condition assessment includes the running carriageway, hard shoulder and other lanes. The Asset Condition entry does not support a direct entry for Cross Section Position (XSP) hence HighStone configurations will be set up with a separate Asset record for each Running Lane. These Asset records are usually used only for the recording of Condition Assessment records.

The list of all Condition Assessments can be obtained using the first SQL script above, but changing the Where condition from:

```
tblRoomComponents.RCItemCont = 0
```

To:

```
tblRoomComponents.RCItemCont = 1
```



For continuous assets, it is likely that each Asset record will carry two or more Assessment Condition defect records as part of each survey - with different conditions being recorded at different points along the asset length. This means that the SQL structure used above cannot be used as multiple records may exist for each Asset Record / Survey Section Record combination.

To overcome this one option is to use a double pass:

- Establish a list of Asset Identifier / Survey Section Identifier combinations that identifies the last Condition Survey undertaken for each Asset record. This can be achieved using the same Row_Number() function to identify each target record.
- Run the final report of Asset Conditions by linking the Condition Assessments (held in [tblDefects]) to the prepared list of Asset Identifier / Survey Section Identifier combinations.

The approach assumes that each Asset Condition Survey completed the full length of each geographic Section Survey. If surveys do not complete a full section length then an alternative approach may be required to give a full report - for instance:

- Listing all survey results completed within a defined time span - e.g. over the last 12 month period.
- Add markers to the underlying Condition Defects records to identify the current 'live' assessment records [this is seem very much as a last resort and casting markers across this large dataset to maintain the required extract will be very inefficient and difficult to maintain].
- Custom reporting that analyses the stored geographic details rigorously [outside the scope of this document].

The following SQL extract prepares a temporary list of Asset Identifier / Survey Section Identifier combinations to seed a final report. This could be set to use the HighStone internal pivot table [tblPivotReport], but here uses a custom temporary table:

```
Insert Into tmpTable (AssetId, SurveySecId)
Select RoomsRCId, DFSurveyId From (
Select tblRoomsRC.RoomsRCId, tblDefects.DFSurveyId, tblDefects.DFDateRaised,
Row Number() Over (Partition By tblRoomsRC.RoomsRCId Order By tblDefects.DFDateRaised DESC) As Rn
From tblRoomsRC Inner Join
tblRoomComponents On tblRoomsRC.RoomsRCCompId = tblRoomComponents.RComponentId
Inner Join
tblDefects On tblRoomsRC.RoomsRCId = tblDefects.DFItemId Inner Join
tblMasterDefectItems On tblDefects.DFClassId =
tblMasterDefectItems.MDIIdentifier
Where tblRoomComponents.RCItemCont = 1 And tblMasterDefectItems.MDICode = 'LGST'
And tblRoomsRC.RoomsRCDeleted = 0
)tTable
Where Rn = 1;
```

The final report can then use this temporary table as a feed for the final data report:

```
Select tblRoomsRC.RoomsRCId, tblRoomComponents.RCItemCode, tblRooms.RoomNumber,
ltbXSP.XSPTagCode, tblRoomsRC.RoomsRCStChain, tblRoomsRC.RoomsRCFnChain,
tblDefectPriority.DPDescription, tblDefects.DFDateRaised,
tblProjectSurveys.PSSurveyNo
From tmpTable Inner Join
tblDefects On tmpTable.PVHeadingId = tblDefects.SurveySecId And
tmpTable.AssetId = tblDefects.DFItemId Inner Join
tblDefectPriority On tblDefects.DFCategory = tblDefectPriority.DPIIdentifier
Inner Join
tblMasterDefectItems On tblDefects.DFClassId =
```



```
tblMasterDefectItems.MDIIdentifier Inner Join
tblSectionSurveys On tblDefects.DFSurveyId = tblSectionSurveys.SecSurveyId
Inner Join
tblProjectSurveys On tblSectionSurveys.SSSurveyId = tblProjectSurveys.SurveyId
Inner Join
tblRoomsRC On tblDefects.DFItemId = tblRoomsRC.RoomsRCId Inner Join
tblRoomComponents On tblRoomsRC.RoomsRCCompId = tblRoomComponents.RComponentId
Inner Join
tblRooms On tblRoomsRC.RoomsRCRoomId = tblRooms.RoomId Left Join
ltbXSP On tblRoomsRC.RoomsRCXRefCodeId = ltbXSP.XSPId
Where tblMasterDefectItems.MDICode = 'LGST'
Order By tblRoomsRC.RoomsRCId, tblRoomsRC.RoomsRCStChain;
```

If preferred, this twin script approach could be run across both Continuous and Point Asset items in a single pass to give a complete data listing. However, it is usual to run Condition Reports against different asset classes rather than call for entire contract wide data extracts.

Asset Secondary Conditions

Certain Assets may require more than a single Condition Assessment to reflect the characteristics of the asset type. A couple of examples that have been identified are:

- Road Signage - Condition assessments for the sign plate, and the supporting column if appropriate.
- Lighting Columns - Condition assessments for different elements of the item - e.g. column and lantern.

Exactly what level of detail is to be collected during surveys is down to the user to consider and define. It is recommended that requirements are always considered against the absolute requirement to collect relevant information, and also consider the practicality of both data collation and maintenance.

There are two options to compiling data to meet these requirements:

- Split the elements of the Asset (e.g. Sign Plate and Supporting Column(s)) in to separate Asset Item types and manage as record items in their own right. Whilst a logical approach from a data point of view, it is understandable that the additional record processing is undesirable.
- Create a new Asset Condition entry as part of the generic 'Defect' class definitions. This provides the facility for inspectors to make twin assessments against the asset records.

As this need for separate Secondary Condition applies to only certain Asset Types, it is recommended that the following implementation is made in HighStone:

- Create a parallel entry in the parent Master Defect classification table [tblMasterDefects]. This addition will sit alongside the standard entry for 'Asset Condition' already defined. The addition is made here as the entry can be targeted to apply to only specified asset types (and not across all asset record types).
- A child entry must be made in the item definition for Defect Type table [tblMasterDefectItems]. This record must be linked back to the parent record, and can include a suitable caption.

The new entry will be applied as a 'Condition' class entry and the user will enter the assessment against the options of 'As New', 'Satisfactory', 'Poor' and 'Failed' as defined in [tblDefectPriority].

Asset Condition reports may need to be adjusted to include the new Asset Defect Class definitions.



Some applications including HighStone, HighField and other data capture programs may require changes to coding and / or configuration to support new Asset Condition reporting definitions.

The Asset Condition recording held in HighStone as part of the standard asset inspections is designed to hold basic assessments as to the condition of each asset item. Technical asset items such as electrical circuits, electrical lamps, are likely to be subjected to a more rigorous regime of inspection and test that collates much more detailed information. Details on this level of testing will be held in other data definitions within the Contract Database.

Asset Cyclic Maintenance Tasks

Assets will be subjected to routine maintenance and cleaning tasks and details on when these tasks are undertaken will usually need to be recorded. In creating the definitions to hold this information, the level of detail that is to be recorded must be decided, as follows:

- Given the geographic Section definitions, cyclic maintenance may be recorded as a completed task on a section by section basis. Thus tasks such as litter picking, sweeping can be recorded on the basis of a schedule that rotates through all sections on the Contract Network. If this approach is taken, then reporting maintenance activities against individual asset records will not be possible.
- Alternatively cyclic maintenance may be recorded against individual assets held in the Contract Database. This option involves more data entry and management - particularly where the Contract Network includes many assets such as Lamps (Road Lighting) and Gullies (Drains). However, this structure does allow a full history of Defects, Condition Assessments and Cyclic Maintenance to be given on an individual asset basis.

The user may also wish to consider how non-cyclic activities are to be handled and recorded, for example the replacement of assets brought about by accident damage or unit failure. Whilst the overall upgrade / renewal of a whole section of assets as part of long term asset renewal would be expected to lead to the introduction of new Asset Records to the Contract Database (and hence the start of a new asset maintenance history), ad hoc replacements on a like-for-like basis would not normally initiate a new Asset Record. If such data is to be collated under the Asset Maintenance records then each type of maintenance activity that is to be recorded must be defined in the Contract Database.

The following examples give some guidance on what could be considered:

Grass Areas / Verges	Swathe Grass Cut Visibility Grass Cut
Horticulture	Shrub Pruning Tree Pruning Branch Trimming
Hardstand / Carriageway Areas	Sweeping
Verges / Central Reserve	Litter Picking
Drainage	Inspection / Dipping Gully Cleaning Channel Cleaning Slotted Channel Cleaning



Signage	Sign Plate Cleaning Sign Plate Replacement Column Replacement
Lamps / Lighting	Lantern Cleaned Lantern Replaced Lamp Replacement Column Painted Lamp Removed (Damaged) Lamp Installed Lamp On OK ¹ Lamp Out ¹ Lamp Missing ¹

¹ - These classifications are designed to carry the results of routine Lamp Scouts run at night to check that lighting is fully working.

In managing data records entered through the generic 'Asset Defects' records, it is important to understand and maintain the link to the Network Surveys structures. It is a requirement of HighStone that all record entries in [tblDefects] are linked to relevant entries in the Surveys tables [tblProjectSurveys] and [tblSectionSurveys]. In many cases these Survey record entries are created automatically by HighStone when asset maintenance details are entered, but in some instances user will need to raise a specific Survey record in order to complete Cyclic Maintenance entries.

Simplified Asset Maintenance Records

The above definitions are the preferred data structures for HighStone use as the information collected builds in to a full Asset Maintenance history for each asset record. External asset registers typically carry details of cyclic maintenance alongside definitions for the asset characteristics - typically in a flat style spreadsheet or data record. Such elements usually include:

- Date of installation
- Date last inspection
- Condition of Asset at last survey
- Date of last clean
- Date last lamp change
- Note about asset

In bringing this information in to HighStone data structures, users may consider holding this type of data as part of the Asset Attribute characteristics (holding data values within the data table [tblRoomsCA]). HighStone will support such data configuration however it is not a recommended approach.

Entries held as part of the Asset Attributes are designed to hold just a single version of the data value and cannot carry a series of changing values. Thus such entries can carry *only* the 'Date of Last Survey' - and once the value is changed to reflect a more recent survey the date reference to



the previous (and earlier) Surveys is lost. Similar issues relate to holding details such as Asset Condition, and any Notes taken in the field. The wider HighStone data definition for Surveys and Defects records overcome this restriction, and hence are strongly recommended.

Holding such data values directly as part of the Asset Attribute definitions can make user access to the values easier and simplify the creation of some reporting functions.

If the user does set Asset Attribute entries to carry this type of data values against assets records the following points should be followed:

- Master Asset Attribute definitions must be declared for each Attribute Type in data table [tblComponentAttribs].
- For each attribute type that is declared, the actual attribute data value will be held as a record in the data table [tblRoomsCA].
- Users will have to maintain the current data value for each attribute, for each asset record, manually. HighStone does not include functions on the main display panels to update this style of asset data definition.
- As an alternative, the asset attribute values could be refreshed periodically to reflect changes logged against recent Survey Records. The SQL scripts given above could be used to establish the date of last survey, or last recorded condition, for an asset record and cast this against the target Asset Attribute record. The HighStone SQL construction **BLOCK RECAST Select ... APPLY AS Update ...** would achieve this casting.

A certain level of data management on the part of the user will be necessary to ensure a consistent data set is maintained at all times.